The University of Warwick

THEORY OF COMPUTATION

REPORT NO. 64

Some Results On Replacement Rules in Monotone Boolean Networks

by

Paul E Dunne

University of Warwick
Department of Computer Science
Coventry CV4 7AL
England

January 1984

Some Results On Replacement Rules In Monotone Boolean Networks

Paul. E. Dunne

Department Of Computer Science
University Of Warwick
Coventry CV4 7AL
Great Britain

1. Introduction

Replacement rules were introduced by Paterson [4] and Mehlhorn [3] and used to prove tight lower bounds on the monotone network complexity of boolean matrix multiplication. The results applied prove that in networks computing boolean matrix product, gates computing certain functions may be replaced by the constants 0 or 1 or by an input of the network. Applications of replacement rules to other functions have been made by Weiss [6] for boolean convolution and Dunne [2] for threshold functions.

In this paper we investigate the following problem:

(P1) Given a pair of boolean functions $f:\{0,1\}^n\to\{0,1\}$, $g:\{0,1\}^n\to\{0,1\}$ what are the monotone boolean functions h such that for any monotone network $\mathbf{S}$ computing f, containing a gate $\mathbf{u}$ which computes the function g, $\mathbf{S}^{|RES(\mathbf{u}):=h}$ still computes f?

We prove the following results:

R1) For any function f we derive closed form expressions for the maximal 0-replaceable and minimal 1-replaceable functions with respect to f.

R2) For any pair of functions f, g we determine closed form expressions for:

(i) min s such that g is replaceable by s in a network computing f

(ii) max s such that g is replaceable by s in a network computing f

R3) For any pair of functions f,g we determine closed form expressions for:

(i) min s such that s is replaceable by g in a network computing f

(ii) max s such that s is replaceable by g in a network computing f

Using (R2) we obtain a complete solution for (P1).

The remainder of this paper is organised as follows. Below we give basic definitions and the notation used. In Section(2) we derive (R1), (R2) and (R3) above. In Section(3) we extend our results to multiple-output functions and illustrate some applications. In Section(4) we prove the existence of "pseudo-complements" for all single-valued monotone boolean functions and characterise these.

Definition 1

A *monotone network* **S**, is a directed acyclic graph with 2 distinguished sets of nodes: **X** (the *inputs* of **S**) is the set of nodes with in-degree 0. These nodes are labelled with members of the set $\{x_1,\ldots,x_n\}$. **G** is the remaining set of nodes, which have in-degree 2 (the *gates* of **S**). Gates are labelled by $\wedge$ or $\vee$ (Boolean conjunction and disjunction).

□

Definition 2

If **S** is a monotone network and **u** is a node of **S**, $RES(\mathbf{u})$ is the boolean function recursively defined by:

$$RES(\mathbf{u}) = \begin{cases} x_i \text{ if } \mathbf{u} \text{ is input } x_i \text{ of } \mathbf{S} \\ RES(\mathbf{u}_1) \wedge RES(\mathbf{u}_2) \text{ if } \mathbf{u} \text{ is an } \wedge \text{ gate} \\ RES(\mathbf{u}_1) \vee RES(\mathbf{u}_2) \text{ if } \mathbf{u} \text{ is an } \vee \text{ gate} \end{cases}$$

where $\mathbf{u}_1$ and $\mathbf{u}_2$ are the inputs of **u** if **u** is a gate. □

Definition 3

$$f(X) \leq g(X) <=> \forall\, \alpha \in \{0,1\}^n\, f(\alpha)=1 => g(\alpha)=1$$

□

Definition 4

A monom **m** is a function of the form:

$\mathbf{m} = x_{i\,1} \wedge x_{i\,2} \wedge \ldots \wedge x_{i\,q} \quad x_{i\,j} \in \mathbf{X}$

A monom **m** is an *implicant* of the monotone boolean function f if and only if $\mathbf{m} \leq f$. **m** is a *prime implicant* if:

1) **m** is an implicant of f

2) $\forall$ $\mathbf{m}'$ such that $\mathbf{m} < \mathbf{m}'$, $\mathbf{m}'$ is not an implicant of f

□

Definition 5

A *clause* $\mathbf{c}$ is a function of the form:

$\mathbf{c} = x_{i1} \vee x_{i2} \vee \ldots \vee x_{iq} \quad x_{ij} \in \mathbf{X}$

A clause $\mathbf{c}$ is an *implicand* of the monotone boolean function f if and only if $f \leq \mathbf{c}$. $\mathbf{c}$ is a *prime clause* if:

1) $\mathbf{c}$ is an implicand of f

2) $\forall\, \mathbf{c}'$ such that $\mathbf{c}' < \mathbf{c}$, $\mathbf{c}'$ is not an implicand of f.

□

Notation

$PI(f) = \{ \mathbf{m} \mid \mathbf{m}$ is a prime implicant of $f \}$

$PC(f) = \{ \mathbf{c} \mid \mathbf{c}$ is a prime clause of $f \}$

2. Results

Definition 6

Let $f:\{0,1\}^n \to \{0,1\}$, $g:\{0,1\}^n \to \{0,1\}$ and $h:\{0,1\}^n \to \{0,1\}$ be any monotone boolean functions.

g is *h-replaceable with respect to* f ($g \overset{f}{=|} h$) iff:

$$\mathbf{S}^{|RES(\mathbf{u}):=h} \quad \text{still computes } f$$

for any monotone network $\mathbf{S}$ computing f which contains a node $\mathbf{u}$ with $RES(\mathbf{u})=g$ □

The following result is the replacement rule due to Mehlhorn [3].

Lemma 1

$g \overset{f}{=|} 0$ iff:

$\forall\, \mathbf{m} \in \mathbf{PI}(g)$ $\neg\exists\, \mathbf{m}'$ such that

$$\mathbf{m} \wedge \mathbf{m}' \in \mathbf{PI}(f)$$

□

An easy corollary of this is:

Lemma 2

$g \overset{f}{=|} 1$ iff:

$$\forall h \quad g \wedge h \leq f \;\Rightarrow\; h \leq f$$

□

We shall now characterise the largest 0-replaceable and smallest 1-replaceable functions with respect to any function f. Although the results are implied by Theorem(3) below, we shall present these cases separately as the analysis is clearer. The characterisations use the representation of f as a conjunction of prime clauses (Conjunctive Normal Form) or as a disjunction of prime implicants (Disjunctive Normal Form).

Definition 7

Let f be a monotone boolean function over $\{x_1,\ldots,x_n\}$. The *dual* of f $(\bar{f})$ is the monotone function:

$$\bar{f} = \neg (f (\neg x_1,\ldots, \neg x_n))$$

where "$\neg$" denotes negation.

□

Definition 8

Let $\mathbf{m}$ be a monom, $\mathbf{c}$ be a clause and f be a monotone boolean function. Then:

1) $\chi(\mathbf{m}) = \vee \{ x \mid \mathbf{m} \not\leq x \}$

2) $OC(f) = \bigwedge_{\mathbf{m} \in PI(f)} \chi(\mathbf{m})$

3) $\varphi(\mathbf{c}) = \wedge \{ x \mid x \not\leq \mathbf{c} \}$

4) $IC(f) = \bigvee_{c \in PC(f)} \varphi(c)$

□

Theorem 1 (0,1-replacements)

(A) $OC(f)$ is the unique largest 0-replaceable function with respect to f. i.e

$OC(f) < g \Rightarrow g$ is not 0-replaceable w.r.t f.

(B) $IC(f)$ is the unique smallest 1-replaceable function with respect to f. i.e.

$IC(f) > g \Rightarrow g$ is not 1-replaceable w.r.t f.

Proof

Let:

$$Dross(f) = \vee \{ g \mid g \overset{f}{=\!|} 0 \}$$

We shall show that:

$$OC(f) = Dross(f)$$

(i) $OC(f) \leq Dross(f)$

Suppose $\mathbf{m} \leq OC(f)$, but that $\mathbf{m} \not\leq Dross(f)$. Then there is some monom $\mathbf{m}'$ such that $\mathbf{m} \wedge \mathbf{m}' \in PI(f)$ and so $\mathbf{m}$ is a shortening of some prime implicant $\mathbf{r}$ of f. But:

$$\mathbf{m} \leq OC(f) = \bigwedge_{p \in PI(f)} \chi(\mathbf{p}) \Rightarrow \mathbf{m} \leq \chi(\mathbf{r})$$

By monotonicity

But $\mathbf{m} \not\leq \chi(\mathbf{r})$ if $\mathbf{m}$ is a shortening of $\mathbf{r}$. Contradiction.

(ii) $Dross(f) \leq OC(f)$

Let $\mathbf{m} \in \mathbf{PI}\ (\ Dross(f)\ )$, by Lemma(1) there does not exist $\mathbf{m}'$ such that $\mathbf{m} \wedge \mathbf{m}' \in \mathbf{PI}(f)$. Thus:

$$\forall\, \mathbf{r} \in \mathbf{PI}(f) \quad \mathbf{m} \leq \chi(\mathbf{r})$$

$$\mathbf{m} \leq \bigwedge_{\mathbf{p} \in \mathbf{PI}(f)} \chi(\mathbf{p}) \leq OC(f)$$

Thus:

$$OC(f) = Dross(f)$$

and (A) follows as $Dross(f)$ is by definition the largest 0-replaceable function with respect to f.

(B) It is easy to see that:

$$IC(f) = \overline{OC(\bar{f})}$$

Thus, by duality, $IC(f)$ is the unique minimal 1-replaceable function with respect to f.

□

We shall now consider non-constant replacements of the form $f{:=}s$, and determine minimum and maximum solutions for these.

Definition 9

Let $\mathbf{M}=\{\mathbf{m}_1,\ldots,\mathbf{m}_k\}$ be a set of monoms, and let f be a monotone boolean function. The *Prime- Implicant Extension* of $\mathbf{M}$ with respect to f ($\mathbf{IE}_f(\mathbf{M})$) is defined to be:

$$\mathbf{IE}_f(\mathbf{M}) = \{\, \mathbf{p} \in \mathbf{PI}(f) \mid \exists\, \mathbf{m}_i \in \mathbf{M} \text{ with } \mathbf{p} \leq \mathbf{m}_i \,\}$$

The *Prime-Clause Extension* of a set of clauses $\mathbf{C} = \{\, \mathbf{c}_1,\ldots,\mathbf{c}_k \,\}$ with respect to f ($\mathbf{CE}_f(\mathbf{C})$) is given by:

$$\mathbf{CE}_f(\mathbf{C}) = \{\, \mathbf{p} \in \mathbf{PC}(f) \mid \exists\, \mathbf{c}_i \in \mathbf{C} \text{ with } \mathbf{c}_i \leq \mathbf{p} \,\}$$

$$A(f,g) = \bigvee_{\mathbf{m} \in \mathbf{IE}_f(\mathbf{PI}(g))} \mathbf{m}$$

$$B(f,g) = \bigwedge_{\mathbf{c} \in \mathbf{CE}_f(\mathbf{PC}(g))} \mathbf{c}$$

□

Theorem 2

Let f, g be monotone boolean functions. Then:

1) $A(f,g)$ is the minimal function s such that $g \overset{f}{=\!|} s$

2) $B(f,g)$ is the maximal s such that $g \overset{f}{=\!|} s$

Note: Conventionally the empty monom (clause) is 1 (0).

Proof

1) Certainly g is $A(f,g)$-replaceable when computing f, as by definition $\mathbf{IE}_f(\mathbf{PI}(g))$ is the set of all prime implicants of f to which g could be extended. So suppose some function s. exists, also satisfying these requirements and that $A(f,g) \not\leq s$ There must be some prime implicant of $A(f,g)$, $\mathbf{p}$ say, which is not an implicant of s. Now:

$\mathbf{p} \in \mathbf{PI}(f)$ and $\exists\, \mathbf{m} \in \mathbf{PI}(g)$ such that $\mathbf{p} \leq \mathbf{m}$

So; $\mathbf{PI}(g \wedge \mathbf{p}) \cap \mathbf{PI}(f) = \{\mathbf{p}\}$

But; $\mathbf{PI}(g \wedge \mathbf{p}\, s) \cap \mathbf{PI}(f) = \{\}$ as $\mathbf{p} \not\leq s$

Contradiction, as g is not replaceable by s when computing f.

Thus $A(f,g)$ is a minimal function. We now establish uniqueness. Suppose s_1, s_2 are distinct i.e incomparable minimal functions. Then:

$$\mathbf{PI}(g \wedge h) \cap \mathbf{PI}(f) \subset \mathbf{PI}(g \wedge s_1 \wedge h) \cap \mathbf{PI}(f)$$

$$\subset \mathbf{PI}(g \wedge s_1 \wedge s_2 \wedge h) \cap \mathbf{PI}(f)$$

$$\text{as } g \wedge s_1 \wedge h \leq f$$

Thus $s_1 \wedge s_2$ $(\leq s_1, s_2)$ is also a suitable, but smaller function. Contradiction.

2) Duality

□

Corollary 2.1)

$g \overset{f}{=\!|} h$ if and only if:

$$\vee \mathbf{IE}_f(g) \leq h \leq \wedge \mathbf{CE}_f(g)$$

□

Definition 10

$\mathbf{PI}_{\mathbf{rem}}(f,g) = \mathbf{PI}(f)\text{-}\mathbf{IE}_f(\mathbf{PI}(g))$

$\mathbf{PC}_{\mathbf{rem}}(f,g) = \mathbf{PC}(f)\text{-}\mathbf{CE}_f(\mathbf{PC}(g))$

$E(f,g) = \bigwedge_{\mathbf{m} \in \mathbf{PI}_{\mathbf{rem}}(f,g)} \chi(\mathbf{m})$

$D(f,g) = \bigvee_{\mathbf{c} \in \mathbf{PC}_{\mathbf{rem}}(f,g)} \varphi(\mathbf{c})$

□

Theorem 3

(A) $E(f,g)$ is the maximal s such that $s \overset{f}{=\!|} g$.

(B) $D(f,g)$ is the minimal s such that $s \overset{f}{=\!|} g$.

Proof

We need only prove (A) as (B) will follow by duality.

1) $E(f,g) \overset{f}{=\!|} g$

By Cor(2.1) we need only show:

$$A(f,E(f,g)) \leq g \leq E(f,g)$$

(i) $g \leq E(f,g)$

Let $\mathbf{p} \leq g$. Then by definition of $\mathbf{PI_{rem}}(f,g)$:

$$\neg\exists \mathbf{m}' \text{ such that } \mathbf{p} \wedge \mathbf{m}' \in \mathbf{PI_{rem}}(f,g)$$

Thus:

$$\forall\, \mathbf{m} \in \mathbf{PI_{rem}}(f,g)\ \ \mathbf{p} \leq \chi(\mathbf{m})$$

By the definition of $E(f,g)$ this implies the result.

(ii) $A(f,E(f,g)) \leq g$

Let: $\mathbf{p} \in \mathbf{PI}(\, A(f,E(f,g))\,)$. Now:

$$\mathbf{IE}_f(\, \mathbf{PI}(E(f,g))\,) \cap \mathbf{IE}_f(\, \mathbf{PI_{rem}}(f,g)\,) = \{\}$$

Thus either $\mathbf{p} \in \mathbf{PI}(f)$, in which case $\mathbf{p}$ is a lengthening of some prime implicant $\mathbf{m}$ of g or $\mathbf{p}=0$. In both cases $\mathbf{p} \leq g$.

2) $E(f,g)$ is maximal

Suppose $\mathbf{u} \not\leq E(f,g)$ and $\mathbf{u} \overset{f}{=\!|} g$. Then

$$\exists\, \mathbf{p} \in \mathbf{PI}(\mathbf{u}) \text{ such that } \mathbf{p} \not\leq E(f,g)$$

Thus: $E(f,g) \leq \chi(\mathbf{p})$ and therefore,

$\exists\, \mathbf{r} \in \mathbf{PI_{rem}}(f,g)$ such that $\chi(\mathbf{r}) \leq \chi(\mathbf{p})$

So $\mathbf{r} \leq \mathbf{p}$. Thus:

$\mathbf{u} \wedge \mathbf{r} = \mathbf{r} \in \mathrm{PI}(f)$

$g \wedge \mathbf{r} \neq \mathbf{r}$ (As $\mathbf{r} \in \mathrm{PI}_{\mathrm{rem}}(f,g)$)

Thus $\mathbf{u}$ is not g-replaceable with respect to f. Contradiction. □

Corollary 3.1)

$h \overset{f}{=\!|} g$ if and only if:

$$\bigvee_{\mathbf{c} \in \mathrm{PC}_{\mathrm{rem}}(f,g)} \varphi(\mathbf{c}) \leq h \leq \bigwedge_{\mathbf{m} \in \mathrm{PI}_{\mathrm{rem}}(f,g)} \chi(\mathbf{m})$$

□

Beynon [1] has considered a concept of "computational equivalence" within a different framework. g is said to be equivalent to h when computing f ($g \overset{f}{\models\!\models} h$) if and only if $g \overset{f}{=\!|} h$ and $h \overset{f}{=\!|} g$. In this context Cor(2.1) and Cor(3.1) yield:

Theorem 4

$g \overset{f}{\models\!\models} h$ iff

$$A(f,g) \vee D(f,g) \leq h \leq B(f,g) \wedge E(f,g)$$

Proof

Obvious

□

3. Multiple Output Functions

Let $F = \{ f_1,\ldots f_m \}$ be any set of m monotone boolean functions over $\mathbf{X}$

Theorem 5

$$OC(F) = \bigwedge_{i=1}^{m} OC(f_i)$$

$$IC(F) = \bigvee_{i=1}^{m} IC(f_i)$$

$$A(F,g) = \vee \mathbf{IE}_F\,(\mathbf{PI}(g))$$

$$B(F,g) = \wedge \mathbf{CE}_F\,(\mathbf{PC}(g))$$

$$E(F,g) = \bigwedge_{i=1}^{m} E(f_i,g)$$

$$D(F,g) = \bigvee_{i=1}^{m} B(f_i,g)$$

where;

$$\mathbf{IE}_F(\mathbf{M}) = \{\mathbf{p} \in \bigcup_{i=1}^{m} \mathbf{PI}(f_i) \mid \exists \mathbf{m}_i \geq \mathbf{p}\}$$

$$\mathbf{CE}_F(\mathbf{C}) = \{\mathbf{r} \in \bigcup_{i=1}^{m} \mathbf{PC}(f_i) \mid \exists \mathbf{c}_i \leq \mathbf{r}\}$$

Proof

Elementary

□

As an illustration we reprove the replacement rule due to Paterson [4] for Boolean Matrix Multiplication. Let:

$BMP\ \{0,1\}^{2n^2}\text{->}\{0,1\}^{n^2}$

where each output c_{ij} is defined by:

$$c_{ij} = \bigvee_{\substack{1\le i\le n \\ 1\le j\le n}} (x_{ik} \wedge y_{kj})$$

Lemma 3 (Paterson [4])

Let $BMP = \{ c_{11},\ldots,c_{nn} \}$, where c_{ij} is as defined above. Let $1 \le i,i' \le n$ $(i \ne i')$ and $1 \le j,j' \le n$ $(j \ne j')$.

3.1) $x_{ik} \vee x_{i'k} \overset{BMP}{=} | \ 1$

3.2) $y_{kj} \vee y_{kj'} \overset{BMP}{=} | \ 1$

3.3) $x_{ik} \vee y_{kj} \overset{BMP}{=} | \ 1$

Proof

$$IC(c_{ij}) = \overline{OC(\overline{c_{ij}})}$$

$$= \overline{OC(\bigwedge_{1\le k\le n} (x_{ik} \vee y_{kj}))}$$

$$= \overline{\bigvee_{\substack{1\le p\ne i\le n \\ 1\le q\le n}} x_{pq} \vee \bigvee_{\substack{1\le r\le n \\ 1\le s\ne j\le n}} y_{rs} \vee c_{ij}}$$

$$= \bigwedge x_{pq} y_{rs} \wedge \overline{c_{ij}}$$

Thus:

$$IC(BMP) = \bigvee_{\substack{1\le i\le n \\ 1\le j\le n}} (\bigwedge x_{pq} y_{rs} \overline{c_{ij}})$$

It is easy to see that for each of the function s in (3.1)-(3.3):

$$IC(BMP) \vee s = s \Rightarrow IC(BMP) \leq s$$

□

A k-slice function of f is a function of the form:

$$f \wedge T_k^n \vee T_{k+1}^n$$

The following is due to Wegener[5].

Lemma 4

If g is a k-slice function then:

$$\forall x_i \in X: \; x_i \overset{g}{=|} \; x_i \wedge T_k^n(X)$$

Proof

Easily derived from Theorem(2) above. □

Corollary

If g is a k-slice function then:

$$\forall x_i \in X: \; x_i \overset{g}{=|} \; x_i \vee T_{k+1}^n(X)$$

Proof

Duality. □

4. Replacement Rules & Pseudo-Complements

Definition 11

Let f be a monotone boolean function over **X**. A *pseudo-complement* for x_i is a monotone boolean function h_i, such that in any $(\wedge,\vee,\neg,)$-network **T** computing f, in which negation is applied only to the inputs, any instance of $\neg x_i$, can be replaced by h_i and the resulting network will still compute f.

□

Theorem 6

$\forall$ monotone f, h_i is a pseudo-complement for x_i if and only if:

$$f^{|x_i:=0} \leq h_i \leq f^{|x_i:=1}$$

Proof

Let f_0 denote the function computed by **T** after some instance, z say, of $\neg x_i$ is replaced by $f^{|x_i=0}$. Similarly let f_1 denote the function computed by **T** after this instance, z, is replaced by $f^{|x_i=1}$. Now since $f_0 \leq f_1$ it is sufficient to prove that:

$$f_1 \leq f \leq f_0$$

In terms of the instance z of $\neg x_i$, **T** computes:

$$g_{000} \vee x_i g_{100} \vee \neg x_i g_{010} \vee z g_{001} \vee \neg x_i z g_{011} \vee x_i z g_{101}$$

where the functions $g_{\alpha\beta\gamma}$ are such that:

1) $\forall\ \mathbf{m} \in \mathrm{PI}(g_{\alpha\beta\gamma})$:

$$(x_i)^{\alpha}\,(\neg x_i)^{\beta}\,(z)^{\gamma} \wedge \mathbf{m}$$

is a monom computed at **T**.

2) **m** does not depend on x_i, $\neg x_i$ or z.

where:

$$(x)^{\delta} = \begin{cases} 1 \text{ if } \delta = 0 \\ x \text{ if } \delta = 1 \end{cases}$$

Clearly:

$$f = g_{000} \vee x_i g_{100} \vee \neg x_i\,(g_{010} \vee g_{001} \vee g_{011})$$

Now let $z := f^{|x_i=0}$ so that $f := f_0$. We must prove $f \leq f_0$. We need only show:

$$\neg x_i g_{001} \vee \neg x_i g_{011} \leq f_0$$

However; $f^{|x_i=0} \wedge g_{001} = g_{001}$ and $\neg x_i f^{|x_i=0} g_{011} = \neg x_i\, g_{011}$ thus $f \leq f_0$.

Now consider the replacement $z := f^{|x_i=1}$. We must show that $f_1 \leq f$ Similarly we need only prove:

$$f^{|x_i=1} g_{001} \vee \neg x_i f^{|x_i=1} g_{011} \vee x_i f^{|x_i=1} g_{101} \leq f$$

But:

$$f^{|x_i=1} \wedge g_{001} = g_{001} \leq f$$

$$\neg x_i \wedge f^{|x_i=1} \wedge g_{011} \leq g_{011} \leq f$$

$$x_i \wedge f^{|x_i=1} \wedge g_{101} \leq f$$

Thus $f_1 \leq f$, and the theorem follows. □

Corollary 6.1)

Let F = $\{f_1,\ldots,f_m\}$ be a set of **m** monotone boolean functions. Then h_i is a pseudo-complement for x_i if and only if:

$$\bigvee_{j=1}^{m} f_j^{|x_i=0} \leq h_i \leq \bigwedge_{j=1}^{m} f_j^{|x_i=1}$$

□

We note that for sets of monotone boolean functions, in general the interval of Corollary(6.1) is not well-defined. However for special cases, such as slice functions, pseudo-complements exist. For slice functions Theorem(6) and Corollary(6.1) combined yield the results of Berkowitz (cited in [5]) for transforming combinational networks to monotone networks efficiently.

5. References

[1] Beynon, M

Replaceability and computational equivalence in finite distributive lattices: Theory Of Computation Report No.61, Univ. Of Warwick, March 1984.

[2] Dunne, P.E.

A 2.5n Lower Bound On The Monotone Network Complexity Of T_3^n, Theory Of Computation Report No.62, Univ Of Warwick, March 1984.

[3] Mehlhorn, K. & Galil, Z.

Monotone Switching Networks and Boolean Matrix Product, Computing (16), 99-111, 1976

[4] Paterson, M.S.

Complexity Of Monotone Networks For Boolean Matrix Product, Theoretical Computer Science (1), 13-20, 1975

[5] Wegener, I

On The Complexity Of Slice-Functions, Internal Report, Univ. Of Frankfurt, July 1983

[6] Weiss, J

A $n^{3/2}$ Lower Bound On The Boolean Convolution, Univ. Of Bielefeld, West Germany, November 1982